Critical Thinking Assignment: Rainfall Calculation and Customer Purchase Points Program

# Pseudocode

## Part 1: Pseudocode for Rainfall Calculation

BEGIN

FUNCTION get\_num\_years():

WHILE True:

TRY to get user input as an integer

IF input is less than or equal to 0:

PRINT error message

ELSE:

RETURN number of years

IF ValueError occurs:

PRINT error message

FUNCTION get\_rainfall\_data(num\_years):

INITIALIZE total\_rainfall to 0

CALCULATE total\_months = num\_years \* 12

FOR each year in range(1, num\_years + 1):

PRINT the current year

FOR each month in range(1, 13):

WHILE True:

TRY to get the rainfall for the current month as float

IF rainfall is negative:

PRINT error message

ELSE:

ADD rainfall to total\_rainfall

BREAK the inner loop

IF ValueError occurs:

PRINT error message

RETURN total\_rainfall and total\_months

FUNCTION display\_results(total\_rainfall, total\_months):

CALCULATE avg\_rainfall\_per\_month = total\_rainfall / total\_months

PRINT total\_months, total\_rainfall, and avg\_rainfall\_per\_month

FUNCTION main():

CALL get\_num\_years to get the number of years

CALL get\_rainfall\_data to get the total rainfall and months

CALL display\_results to display the results

CALL main to run the program

END

## Part 2: Pseudocode for Customer Purchase Points

BEGIN

FUNCTION get\_purchase\_count():

WHILE True:

TRY to get user input as integer

IF input is negative:

PRINT error message

ELSE:

RETURN the input value

IF ValueError occurs:

PRINT error message

CLASS Customer:

FUNCTION \_\_init\_\_():

INITIALIZE book\_purchases and points

FUNCTION calculate\_points():

IF book\_purchases >= 8:

SET points to 60

ELSE IF book\_purchases >= 6:

SET points to 30

ELSE IF book\_purchases >= 4:

SET points to 15

ELSE IF book\_purchases >= 2:

SET points to 5

ELSE:

SET points to 0

RETURN points

FUNCTION update\_purchases(purchases):

UPDATE book\_purchases with input

CALL calculate\_points()

FUNCTION display\_points():

PRINT the number of books purchased and the points earned

FUNCTION main():

CALL get\_purchase\_count to get the number of purchases

CREATE Customer object

UPDATE the customer object with the purchases

DISPLAY the points earned

END

# Source Code

Below is the Python source code that implements the Rainfall Calculation and Customer Points Purchase functionality.

## # Part 1: Rainfall Calculation

def get\_num\_years():

while True:

try:

num\_years = int(input('Enter the number of years: '))

if num\_years <= 0:

print("Number of years must be greater than 0. Please try again.")

else:

return num\_years

except ValueError:

print("Invalid input. Please enter an integer greater than 0.")

def get\_rainfall\_data(num\_years):

total\_rainfall = 0

total\_months = num\_years \* 12

for year in range(1, num\_years + 1):

print(f"\nYear {year}")

for month in range(1, 13):

while True:

try:

rainfall = float(input(f"Enter the rainfall (in inches) for month {month}: "))

if rainfall < 0:

print("Rainfall cannot be negative. Please try again.")

else:

total\_rainfall += rainfall

break

except ValueError:

print("Invalid input. Please enter a valid number.")

return total\_rainfall, total\_months

def display\_results(total\_rainfall, total\_months):

avg\_rainfall\_per\_month = total\_rainfall / total\_months

print(f"\nTotal number of months: {total\_months}")

print(f"Total inches of rainfall: {total\_rainfall:.2f}")

print(f"Average rainfall per month: {avg\_rainfall\_per\_month:.2f}")

def main():

num\_years = get\_num\_years()

total\_rainfall, total\_months = get\_rainfall\_data(num\_years)

display\_results(total\_rainfall, total\_months)

if \_\_name\_\_ == '\_\_main\_\_':

main()  
  
# Part 2: Customer Purchase Points  
  
class Customer:

def \_\_init\_\_(self, book\_purchases=0):

self.book\_purchases = book\_purchases

self.points = 0

def calculate\_points(self):

if self.book\_purchases >= 8:

self.points = 60

elif self.book\_purchases >= 6:

self.points = 30

elif self.book\_purchases >= 4:

self.points = 15

elif self.book\_purchases >= 2:

self.points = 5

else:

self.points = 0

return self.points

def update\_purchases(self, purchases):

self.book\_purchases = purchases

self.calculate\_points()

def display\_points(self):

if self.book\_purchases < 0:

print(f'Verify amount of books purchased and try again, you entered {self.book\_purchases}.')

else:

print(f"You purchased {self.book\_purchases} books and earned {self.points} points this month.")

from Customer import Customer

def get\_purchase\_count():

while True:

try:

purchases = int(input("Enter the number of books you purchased this month: "))

if purchases < 0:

print("Please enter a non-negative number.")

else:

return purchases

except ValueError:

print(f'{purchases} is invalid, please enter a numerical digit equal to or larger than zero: ')

def main():

purchases = get\_purchase\_count()

customer = Customer()

customer.update\_purchases(purchases)

customer.display\_points()

if \_\_name\_\_ == "\_\_main\_\_":

main()

# Screenshots of Execution

## # Part 1: Rainfall Calculation

A screenshot of a computer

Description automatically generated

## # Part 2: Customer Purchase Points

A screenshot of a computer program

Description automatically generated

# Results

The program was executed successfully with multiple test inputs to validate its functionality. In the Rainfall Calculation module, the program effectively captured and processed multi-year rainfall data, computing the total and average rainfall accurately for each year input by the user. The results displayed were clear and formatted appropriately, showing the cumulative rainfall and average monthly rainfall. For the Customer Purchase Points module, the program correctly calculated the reward points based on the number of books purchased, providing immediate and accurate feedback to the user for each purchase scenario. It managed user inputs robustly, including error handling for invalid entries. The outcomes for both modules confirmed their reliability and accuracy, performing as expected under all tested conditions.

# Git Repository

Repository for this week’s assignment can be found here: [GitHub Repository Link for Module 5 Assignment](https://github.com/Mr-Abe/critical_thinking_assignments/tree/master/Module_5_Assignment).